CLAIMS

What is claimed is:

1. A queue structure comprising:

a queue entry pool comprising a plurality of fixed registers configured to store requests wherein each request has a corresponding index; and

an index shifter coupled to the queue entry pool and comprising a plurality of shift registers, wherein each shift register corresponds to one of the plurality of fixed registers and wherein each of the shift registers is configured to store an index corresponding to a request stored in one of the plurality of fixed registers.

2. The queue structure, as set forth in claim 1, wherein the queue entry pool is configured to store read requests.

3. The queue structure, as set forth in claim 1, wherein each of the plurality of fixed registers is configured to store a plurality of flags corresponding to the status of each request.

4. The queue structure, as set forth in claim 3, wherein each of the plurality of fixed registers is configured to store each of a valid entry flag, an entry ready flag and a retire entry flag for the corresponding request.

5. The queue structure, as set forth in claim 4, comprising a flag multiplexor coupled to each of the queue entry pool and the index shifter and configured to receive a plurality of input signals, each of the plurality of input signals corresponding to one of the valid entry flag, the entry ready flag and the retire entry flag and further configured to produce a plurality of output signals each of the plurality of output signals corresponding to one of the valid entry flag, the entry read flag and the retire entry flag.

6. The queue structure, as set forth in claim 5, comprising a priority encoder coupled to each of the flag multiplexor and the queue entry pool and configured to receive each of the plurality of output signals and configured to select one of the requests for execution based on the plurality of output signals.

7. The queue structure, as set forth in claim 1, comprising a queue structure depth checker coupled to the queue entry pool and configured to track the number of requests stored in the queue.

8. A memory controller comprising:

a plurality of processor controller interfaces, wherein each of the plurality of processor controller interfaces is configured to receive requests from one of a processor bus and an input/output bus, each request having a corresponding request type; and

a plurality of queues coupled to each of the processor controller interfaces and configured to store the requests, wherein each request is delivered to one of the plurality of queues depending on the origin of the request and the request type, and wherein each of the plurality of queues comprises:

a queue entry pool comprising a plurality of fixed registers configured to store requests wherein each request has a corresponding index; and

an index shifter coupled to the queue entry pool and comprising a plurality of shift registers, wherein each shift register corresponds to one of the plurality of fixed registers and wherein each of the shift registers is configured to store an index corresponding to a request stored in one of the plurality of fixed registers.

9. The memory controller, as set forth in claim 8, wherein the plurality of queues comprises a plurality of read queues configured to store read requests.

10. The memory controller, as set forth in claim 8, wherein the plurality of queues comprises a plurality of write queues configured to store write requests.

11. The memory controller, as set forth in claim 8, wherein the plurality of queues comprises a plurality of IRV queues configured to store requests associated with a hot-plug event.

12. The memory controller, as set forth in claim 11, wherein the plurality of IRV queues are configured to store each of initialization requests, rebuild requests and verify requests, wherein each of the requests are generated in response to a hot-plug event.

13. The memory controller, as set forth in claim 8, wherein each of the plurality of fixed registers is configured to store a plurality of flags corresponding to the status of each request.

14. The memory controller, as set forth in claim 13, wherein each of the plurality of fixed registers is configured to store each of the valid entry flag, an entry ready flag and a retire entry flag for the corresponding request.

15. The memory controller, as set forth in claim 14, comprising a flag multiplexor coupled to each of the queue entry pool and the index shifter configured to receive a plurality of input signals, each of the plurality of input signals corresponding to one of the valid entry flag, the entry ready flag and the retire entry flag and further configured to produce a plurality of output signals each of the plurality of output signals corresponding to one of the valid entry flag, the entry read flag and the retire entry flag.

16. The memory controller, as set forth in claim 15, comprising a priority encoder coupled to each of the flag multiplexor and the queue entry pool and configured to receive each of the plurality of output signals and configured to select one of the requests for execution based on the plurality of output signals.

17. The memory controller, as set forth in claim 8, comprising a bypass block coupled to each of the plurality of processor controller interfaces and configured to facilitate the execution of the requests received by the plurality of processor controller interfaces without storing the requests in one of the plurality of queues.

18. The memory controller, as set forth in claim 8, comprising a control block coupled to each of the plurality of queues and configured to provide control signals to facilitate the storing and execution of the requests in the plurality of queues.

19. The memory controller, as set forth in claim 18, comprising a control interface block coupled to the control block and further coupled to each of the plurality of processor controller interfaces, wherein the control interface block is configured to transfer the requests from the plurality of processor controller interfaces to the control block.

20. A system comprising:

one or more processors; and

a memory controller coupled to the one or more processors and comprising:

a plurality of processor controller interfaces, wherein each of the plurality of processor controller interfaces is configured to receive requests from one of a processor bus and an input/output bus, each request having a corresponding request type; and

a plurality of queues coupled to each of the processor controller interfaces and configured to store the requests, wherein each request is delivered to one of the plurality of queues depending on the origin of the request and the request type, and wherein each of the plurality of queues comprises:

a queue entry pool comprising a plurality of fixed registers configured to store requests wherein each request has a corresponding index; and

an index shifter coupled to the queue entry pool and comprising a plurality of shift registers, wherein each shift register corresponds to one of the plurality of fixed registers and wherein each of the shift registers is configured to store an index corresponding to a request stored in one of the plurality of fixed registers.

21. The system, as set forth in claim 20, wherein the plurality of queues comprises a plurality of read queues configured to store read requests.

22. The system, as set forth in claim 20, wherein the plurality of queues comprises a plurality of write queues configured to store write requests.

23. The system, as set forth in claim 20, wherein the plurality of queues comprises a plurality of IRV queues configured to store requests associated with a hot-plug event.

24. The system, as set forth in claim 23, wherein the plurality of IRV queues are configured to store each of initialization requests, rebuild requests and verify requests, wherein each of the requests are generated in response to a hot-plug event.

25. The system, as set forth in claim 20, wherein each of the plurality of fixed registers is configured to store a plurality of flags corresponding to the status of each request.

26. The system, as set forth in claim 25, wherein each of the plurality of fixed registers is configured to store each of the valid entry flag, an entry ready flag and a retire entry flag for the corresponding request.

27. The system, as set forth in claim 26, comprising a flag multiplexor coupled to each of the queue entry pool and the index shifter configured to receive a plurality of input signals, each of the plurality of input signals corresponding to one of the valid entry flag, the entry ready flag and the retire entry flag and further configured to produce a plurality of output signals each of the plurality of output signals corresponding to one of the valid entry flag, the entry read flag and the retire entry flag.

28. The system, as set forth in claim 27, comprising a priority encoder coupled to each of the flag multiplexor and the queue entry pool and configured to receive each of the plurality of output signals and configured to select one of the requests for execution based on the plurality of output signals.

29. The system, as set forth in claim 20, comprising a bypass block coupled to each of the plurality of processor controller interfaces and configured to facilitate the execution of the requests received by the plurality of processor controller interfaces without storing the requests in one of the plurality of queues.

30. The system, as set forth in claim 20, comprising a control block coupled to each of the plurality of queues and configured to provide control signals to facilitate the storing and execution of the requests in the plurality of queues.

31. The system, as set forth in claim 30, comprising a control interface block coupled to the control block and further coupled to each of the plurality of processor controller interfaces, wherein the control interface block is configured to transfer the requests from the plurality of processor controller interfaces to the control block.

32. The system, as set forth in claim 20, wherein the system comprises a computer system.

33. The system, as set forth in claim 20, wherein the system comprises a network of computers.

34. A method of processing requests comprising the acts of:

storing requests in a respective fixed register, each request having a corresponding index;

storing the index corresponding to each of the requests in a respective shift register; and

processing the requests in an order determined by the location of each index in the shift register.

35. The method of processing requests, as set forth in claim 34, comprising the act of storing a plurality of flags associated with each request in the respective fixed register, wherein each of the plurality of flags has a corresponding state.

36. The method of processing requests, as set forth in claim 35, comprising the act of reading the corresponding state of each flag to determine a processing order of the requests.

37. The method of processing requests comprising the acts of:

storing a first request in a fixed register, the first request comprising a first index and a first plurality of flags;

storing a second request in the fixed register, the second request comprising a second index and a second plurality of flags;

storing the first index in a first shift register in an entry shifter, wherein the entry shifter comprises a head and a tail;

storing the second index in a second shift register in the entry shifter, wherein the second shift register is closer to the head of the entry shifter than the first shift register;

checking the first plurality of flags to determine whether the first request is ready to be executed;

executing the first request if the first request is ready to be executed;

shifting the second index into the first shift register, if the first request is executed;

checking the second plurality of flags to determine whether the second request is ready to be executed; and

executing the second request if the second request is ready to be executed.

38. The method of processing requests, as set forth in claim 37, wherein the act of storing a first request comprises the act of storing a first read request in a fixed register.

39. The method of processing requests, as set forth in claim 38, wherein the act of storing a second request comprises the act of storing a second read request in a fixed register.

40. A queue structure comprising:

a queue entry pool comprising a plurality of fixed registers configured to store requests; and

an entry shifter coupled to the queue entry pool and comprising a plurality of shift registers, each of the shift registers is configured to store a request corresponding to a request stored in one of the plurality of fixed registers.

41. The queue structure, as set forth in claim 40, wherein the queue entry pool is configured to store read requests.

42. The queue structure, as set forth in claim 40, wherein each of the plurality of fixed registers is configured to store a plurality of flags corresponding to the status of each request.

43. The queue structure, as set forth in claim 42, wherein each of the plurality of fixed registers is configured to store each of a valid entry flag, an entry ready flag and a retire entry flag for the corresponding request.

44. The queue structure, as set forth in claim 43, comprising a flag multiplexor coupled to each of the queue entry pool and the entry shifter and configured to receive a plurality of input signals, each of the plurality of input signals corresponding to one of the valid entry flag, the entry ready flag and the retire entry flag and further configured to produce a plurality of output signals each of the plurality of output signals corresponding to one of the valid entry flag, the entry read flag and the retire entry flag.

45. The queue structure, as set forth in claim 44, comprising a priority encoder coupled to each of the flag multiplexor and the queue entry pool and configured to receive each of the plurality of output signals and configured to select one of the requests for execution based on the plurality of output signals.

46. The queue structure, as set forth in claim 40, comprising a queue structure depth checker coupled to the queue entry pool and configured to track the number of requests stored in the queue.